

AFRL-IF-RS-TR-2005-345
Final Technical Report
September 2005



MAP BASED INTERFACE FOR GEOSPATIAL PRODUCT LIBRARY (MAP-GPL)

LPA Systems, Incorporated

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

**AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK**

STINFO FINAL REPORT

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2005-345 has been reviewed and is approved for publication

APPROVED: /s/

FREDERICK W. RAHRIG
Project Engineer

FOR THE DIRECTOR: /s/

JOSEPH CAMERA, Chief
Information & Intelligence Exploitation Division
Information Directorate

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 074-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE SEPTEMBER 2005	3. REPORT TYPE AND DATES COVERED Final May 04 – Aug 05	
4. TITLE AND SUBTITLE MAP BASED INTERFACE FOR GEOSPATIAL PRODUCT LIBRARY (MAP-GPL)			5. FUNDING NUMBERS C - FA8750-04-C-0197 PE - 63260F PR - 3480 TA - PD WU - 04	
6. AUTHOR(S) David P. Russell, Todd Krokowski and Chris Lawrence				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) LPA Systems, Incorporated 290 Woodcliff Drive Fairport New York 14450			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Research Laboratory/IFEC 525 Brooks Road Rome New York 13441-4505			10. SPONSORING / MONITORING AGENCY REPORT NUMBER AFRL-IF-RS-TR-2005-345	
11. SUPPLEMENTARY NOTES AFRL Project Engineer: Frederick W. Rahrig/IFEC/(315) 330-3906/ Frederick.Rahrig@rl.af.mil				
12a. DISTRIBUTION / AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.				12b. DISTRIBUTION CODE
13. ABSTRACT (Maximum 200 Words) The Map-GPL project set out to create a map based interface for the Augmented Geospatial product library System (AGS). This interface allows the user to interact with the database of geospatial products through a graphical map interface. In the process of creating this interface, a new toolkit was developed. This toolkit provides basic mapping capabilities that can easily be incorporated into a graphical Java application.				
14. SUBJECT TERMS GPL, AGS, Geospatial Product Library, MAP, Software, Library, Java				15. NUMBER OF PAGES 20
				16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL	

Abstract

Location can be incredibly important to understanding information. Understanding the global position of remote sensing data products can be critical to interpreting their results or at least identifying products of interest. With the importance that global position can play, a generic toolkit allowing data products to be viewed in a global context will prove to be invaluable.

The Map-GPL project was conceived primarily to provide a map based selection tool for use with the Geospatial Product Library. Rather than simply develop a tool tailored to only provide a map based interface for GPL, the decision was made to develop a generic toolkit that could be used with AGS or any other application requiring a map interface. Specifically, the project was structured to build a map interface and demonstrate the flexibility of the interface through integrations with the Augmented GPL System and Geo*View.

The map based toolkit, called GeoMap, has been built as an extension of the open source OpenMap system. OpenMap provides tools for layer management and the ability to open a number of geospatial data products. In order to provide the desired capabilities, GeoMap adds additional layer types capable of loading many different image file formats. The GeoIO library used by GeoMap also provides the geospatial coordinate information required to properly locate the image with respect to all other spatially oriented layers.

Having built the basic toolkit, a map based selection tool was built to support the AGS. This selection tool allows the user to select a region of the world to search for image products. Once the system returns a set of available image products, the map based search tool displays the coverage of all of the returned data products in the map interface. The user may choose to view the image coverage as actual image data or merely coverage area.

In addition to the map based selection tool for AGS, the GeoMap toolkit was integrated with Geo*View. In this integration, the GeoMap toolkit simply provides a tool to allow the comparison of geospatial position for a number of different image products. The GeoMap toolkit has wrapped as a Geo*View plugin.

The project has been completely successful in building a reusable geospatial toolkit capable of overlaying many different spatially correlated images and other data products. The toolkit has been designed with a generic interface allowing or integration with any system requiring a map interface for display of geospatially positioned information.

Table of Contents

1	Summary.....	1
2	Introduction	2
3	Methods, Assumptions, and Procedures.....	3
3.1	GeoMap Toolkit Development	4
3.2	AGS Integration Component	5
3.3	Geo*View Integration Component	5
4	Results	7
4.1	GeoMap Toolkit Development	7
4.2	AGS Integration Component	8
4.3	Geo*View Integration.....	9
5	Conclusions.....	11
5.1	Future Work.....	11
6	Recommendations	13
7	References.....	14

List of Figures

Figure 3.1 AGS Map Interface developed with GeoMap	9
Figure 3.2 Geo*View / GeoMap Screenshot	10

List of Tables

Table 2.1 Reused OpenMap Interfaces	4
Table 3.1 Basic GeoMap API Commands	7
Table 3.2 Customized OpenMap Components	8

1 Summary

Location, particularly geospatial location, can be vitally important to understanding information. City planners are not only interested in what types of business and industry their city supports but also where those facilities are located. Image analysts may be able to review satellite and aerial imagery to identify structures and vehicles, but without location that information is essentially useless.

The importance of geospatial location is well understood by the many users of Geographic Information Systems (GIS) used for modern mapping studies. GIS applications provide many capabilities for overlaying and exploring many different types of geospatial information. In many cases, small subsets of the mapping capabilities of a GIS are required for specific applications.

In these applications, a simple toolkit providing basic mapping, georegistration, layer manipulation, and image display capabilities is necessary. The Map-GPL project was conceived to create such a toolkit for use in a variety of applications.

The software toolkit, called GeoMap, was developed to meet this need. GeoMap provides a Java based map canvas for displaying geospatially referenced raster and vector data. In order to speed development, GeoMap was developed as an extension of OpenMap from BBN Technologies. OpenMap is an open source product that performs basic mapping framework.

Par Government Systems Corporation (PGSC) and LPA Systems (LPA) designed the GeoMap toolkit to provide a simple application programming interface (API) for controlling system interactions with the OpenMap framework. In addition to providing this simple API, PGSC created a new software component that allows OpenMap to display images stored in any image format recognized by the GeoIO library developed by PGSC. The GeoIO library provides the same core image access capabilities as the JITC CLEVEL 7 certified Geo*View product available from PGSC.

The results of the integration of GeoIO with OpenMap and the development of the GeoMap API have been excellent. The GeoMap toolkit provides a basis for creation of many different map based tools. In particular, the toolkit has been used as a selection tool for searching the Augmented Geospatial Product Library System (AGS) and a plugin for Geo*View.

GeoMap will continue to provide a readily accessible library for the development of cross-platform Java applications that require basic mapping and geospatial registration capabilities. The toolkit has been demonstrated to be versatile in its use and integration with existing applications. The toolkit should be considered for any development project in the future where basic geospatial location information needs to be manipulated and expressed in an environment that does not require high precision placement and mensuration capabilities.

2 Introduction

This report will briefly describe the development process followed for the creation of the GeoMap software toolkit. This report will summarize and reference other documents created during the project. It is not intended to provide a complete picture of the GeoMap software or its capabilities.

3 Methods, Assumptions, and Procedures

The development of the Map-GPL toolkit, called GeoMap, followed a typical software development process. This process involved requirements analysis, system design, development, and testing.

During the requirements analysis phase, Air Force, PGSC, and LPA representatives met and discussed the needs of systems requiring mapping capabilities. Of particular interest was the AGS system under development at PGSC. In this application, the user has the ability to query a library of geospatial data products to retrieve data items of interest. One of the query types allow the user to select a geographical region of interest. In the AGS prototype, a simple map query interface was developed to provide a basic selection capability. The primary goal of the Map-GPL project was to provide an enhanced map based selection tool for the AGS system.

With this goal in mind, special attention was paid to the recording of requirements that would specifically benefit AGS. These included requirements around image format support to allow the map toolkit to open and display products help in the AGS library. Additional requirements included conversion among differing geospatial coordinate systems, layer transparency requirements, and the ability to interactively choose between display of actual image data and simply a rectangle depicting an image's area of coverage.

Along with specific requirements for the AGS integration, additional requirements were included to provide a more flexible toolkit. Examples of these requirements include the ability of the toolkit to export map products to a standard data format (specifically NITF) and the creation of a generic API to allow integration with other applications beyond AGS.

In addition to the software requirements defined, a set of assumptions were defined. Of the six assumptions stated in the Requirements Analysis Report, only one of the assumptions did not hold. This assumption stated that Geo*View would readily accept a new window type allowing the display of a map interface. While technically the assumption was correct, the decision was made to not incorporate GeoMap directly into the core Geo*View codebase. Instead of a direct integration, the decision was made to integrate GeoMap as a standard Geo*View plugin.

Once the requirements analysis process was complete, a design effort began. This effort defined the API and as well as the inner components that would be developed to meet the requirements previously defined.

Once the design had been defined, a software development plan was created and development of the final software products began.

3.1 GeoMap Toolkit Development

Geospatial mapping frameworks, which provide robust transform and projection capabilities, already exist both freely and commercially. In order to provide a reusable, robust system OpenMap, from BBN Technologies, was chosen as the basis for developing an augmented API. With OpenMap providing a mapping framework, effort was expended to design and develop a larger API to support visualization of Department of Defense (DoD) centric geospatial imagery and enhancing the querying capabilities of Augmented Geospatial Product Library System (AGS).

According to BBN Technologies website (<http://openmap.bbn.com>):

OpenMap™ is a Java Beans™ based toolkit for building applications and applets needing geographic information. Using OpenMap components, you can access data from legacy applications, in-place, in a distributed setting. At its core, OpenMap is a set of Swing components that understand geographic coordinates. These components help you show map data, and help you handle user input events to manipulate that data.

Specifically, several key OpenMap components were identified during the software design period that provide the capabilities required for successful completion of this effort. The resulting product developed using OpenMap components was called the GeoMap Toolkit. Table 3.1 list all reused OpenMap components.

Table 3.1 Reused OpenMap Interfaces

OpenMap Class Name	Description
com.bbn.openmap.Layer	Container to display images and annotations within the OpenMap environment. Created to control how imagery appears at varying spatial scales.
com.bbn.openmap.LayerHandler	Responsible for managing multiple Layers. Provides programmatic functionality to re-order, add, and remove Layers. Since the LayerHandler controls the ordering of the Layers it determines how each individual Layer gets rendered relative to all other Layers.
com.bbn.openmap.MapBean	Map canvas provided to users of OpenMap. Responsible for spatially drawing imagery. The MapBean also provides the core functionality behind the image-space to coordinate-space transformations (and the inverse) feature.

In addition to these components, other components from the PAR Geo*Ware suite were identified at this time to support the visualization of DoD centric imagery; specifically the Geospatial Input Output Library (GeoIO). It was a design goal to not augment or modify GeoIO or OpenMap to provide the GeoMap Toolkit. Details of theses interactions between GeoIO, OpenMap and GeoMap are covered in the Map-GPL Software Design Document.

3.2 AGS Integration Component

The original AGS querying process involved a derivative of the OpenMap software to create a geospatial region-of-interest (ROI). This capability was an improvement from the original process, but additional functionality was needed. The desire for a more robust querying tool facilitated the design of the GeoMap/AGS integration component. Features of a more robust query would also impact the way AGS currently constructs queries. One of the goals of the AGS/GeoMap integration was to support country selection, instead of trying to calculate the geospatial bounds for each country, the AGS metadata would be modified to support queries based on country codes.

The ability to query based upon a country selection was added to AGS using the GeoMap toolkit. Using the underlying shape file representing political boundaries the user can interact with the shape file by clicking on a particular country of interest. A country code associated with the selected country is then sent to AGS. The metadata maintained by AGS is extended to include the relevant country code(s) for a given imagery product. Multiple country selections were incorporated into the integration component as well.

The traditional ROI tool supported multiple, resizable spatial rectangles representing ROIs. Support for querying with multiple ROIs would need to be incorporated into the AGS baseline.

3.3 Geo*View Integration Component

PAR's Geo*View imagery viewer supports multiple imagery formats including imagery with associated geospatial metadata. Aside from viewing imagery, Geo*View allows users to annotate existing imagery products and generate new National Imagery Transmission Format (NITF) products. Currently users are presented with an individual window for each product that is opened within Geo*View. A limitation of Geo*View was that users were not able to spatially align imagery making it difficult to perform an accurate analysis where two or more images were involved. Along with being able to view more than one image in a common perspective, the ability to adjust the layering of overlapping imagery was desired. With layering capabilities Geo*View can load multiple images and examine image similarities without multiple windows. To enhance Geo*View's spatial capabilities the GeoMap Toolkit would be leveraged.

The overall objective of the Geo*View integration component was to demonstrate the robustness of the toolkit. During design, each requirement defined in the Map-GPL Requirements Analysis Report was translated into a functional aspect of the Geo*View integration component. This integration component focused on the visualization portion of the GeoMap toolkit and did not contain tools related to the GeoMap/AGS integration component like the ROI tool or country selection tool.

Users of the GeoMap/Geo*View integration component are presented with annotation and navigation tools similar to those found in Geo*View. Geo*View users can view

multiple products in multiple windows, where as GeoMap users can view multiple products in a single window.

The final objective of the GeoMap/Geo*View integration was to incorporate a saving capability. The saving capability allows users to add imagery to the map canvas, annotations, and/or symbology then save the resulting product in the NITF format. This feature was implemented through a cropping tool in the integration component.

4 Results

Through the development process it was confirmed that a map based toolkit could be effectively developed for use with a number of different applications with different mapping requirements.

GeoMap provides a robust and flexible toolkit for creating simple map interfaces. The toolkit provides a number of basic features ranging from layer management to image file reading. The basic set of commands that would be used in a minimal application of the toolkit are listed in Table 4.1.

Table 4.1 Basic GeoMap API Commands

GeoMap Interface			
GeoMap Object	Method Signature	Return Data Type	Remarks
GeoMapToolkit	getSpatialCanvas()	SpatialCanvas	Returns the map canvas for this toolkit.
GeoMapToolkit	getLayerHandler()	LayerHandler	Returns the LayerHandler object for this toolkit
GeoMapToolkit	generateLayer(File f)	Layer	Generates a Layer that may be added to the LayerHandler from the specified file.
SpatialCanvas	getRendering()	Component	Returns a Component that is the current rendering of the map canvas. Needed by the user to visualize the 2-D map environment.
LayerHandler	add(Layer l)	void	Adds a new Layer to the LayerHandler
LayerHandler	delete(Layer l)	void	Deletes the given Layer from the LayerHandler
LayerHandler	setLayerPosition(Layer l, int p)	void	Sets the position of a specified Layer.
Layer	setVisible(boolean v)	void	Sets the visibility state of a given Layer.
Layer	setTransparency(float trans)	void	Using a floating point value between 0.0 and 1.0, the user of a Layer can set the transparency of the data for a Layer.

In order to demonstrate the capabilities of the GeoMap toolkit, an interface for the AGS has been developed based on GeoMap. The AGS interface provides a mechanism for a user to first select a region of the world to search for geospatial data products. Once the list of available geospatial data products have been returned by the AGS search engine, the GeoMap interface can display the coverage areas for the different geospatial data products.

4.1 GeoMap Toolkit Development

Using the OpenMap mapping framework proved to be beneficial. To suit the needs of the Map-GPL effort, the OpenMap package needed to be customized. The documentation as well as the ad-hoc e-mail support for customizing OpenMap was more than sufficient,

easing the development process. Table 4.1 summarizes which OpenMap components were modified during development of the GeoMap toolkit.

Table 4.2 Customized OpenMap Components

OpenMap Class Name	Alterations and Enhancements
com.bbn.openmap.Layer	The Layer concept was enhanced for optimized rendering. Other alterations include the ability to adjust the transparency of an individual Layer as well as control at what scale factor the Layer gets rendered.
com.bbn.openmap.LayerHandler	The LayerHandler was extended to manage the new instances of GeoMap Layers mentioned above.
com.bbn.openmap.MapBean	A MapBean capable of threaded rendering was created to increase visualization performance.

Support for shape files and the MrSID format were added to GeoIO during the GeoMap development process. Having the ability to overlay shape file products over existing imagery using GeoMap drove the development for shape file support. Currently the GeoMap toolkit is the only medium where shape files can be visualized. Section 5.1 discusses how support for shape files could be enhanced. The MrSID format utilizes a third party rendering engine that interprets the proprietary format. Geo*View and GeoMap users now have the ability to view imagery in the MrSID format using an interface developed to interact with GeoIO and the MrSID rendering engine.

4.2 AGS Integration Component

Implementing country selection was successfully achieved with the GeoMap/AGS integration component. Users have the ability to select individual or multiple countries. Selecting countries involved making use of two separate instances of the OpenMap MapBean component. A layer developed by OpenMap enabling selections of areas defined by an underlying shape file was used to extract an associated country code with a logical shape representing a country. Memory limitations were observed when trying to incorporate this layer with the instance of GeoMap driving the graphical user interface (GUI). Therefore a separate component running in a different memory space was utilized to provide acceptable performance. Relative to the GeoMap/AGS integration, the end result of the country selection process is passing a collection back to AGS where a proper query can be created.

The integration component also provided a collection of spatial ROIs. A mouse driven tool was developed to assist the AGS user in constructing one or more spatial ROIs. Anchors used as points of entry where users could resize and/or move a spatial ROI were drawn on top of each ROI. The tool made use of the layer concept to maintain the number and physical characteristics of the regions.

Mouse driven zoom and pan tools were added to the GUI to increase usefulness. Both tools operate by first selecting the tool and then positioning the tool over the spatial canvas to carry out its specified function. By clicking on an arbitrary region inside of the spatial canvas the zoom tool adjusts the viewable area presented to the user. To use the pan tool, users must click-and-drag over the spatial canvas. When the user releases the mouse button the viewable area is spatially shifted. Figure 4.1 shows an example implementation of the GUI for the GeoMap/AGS integration.

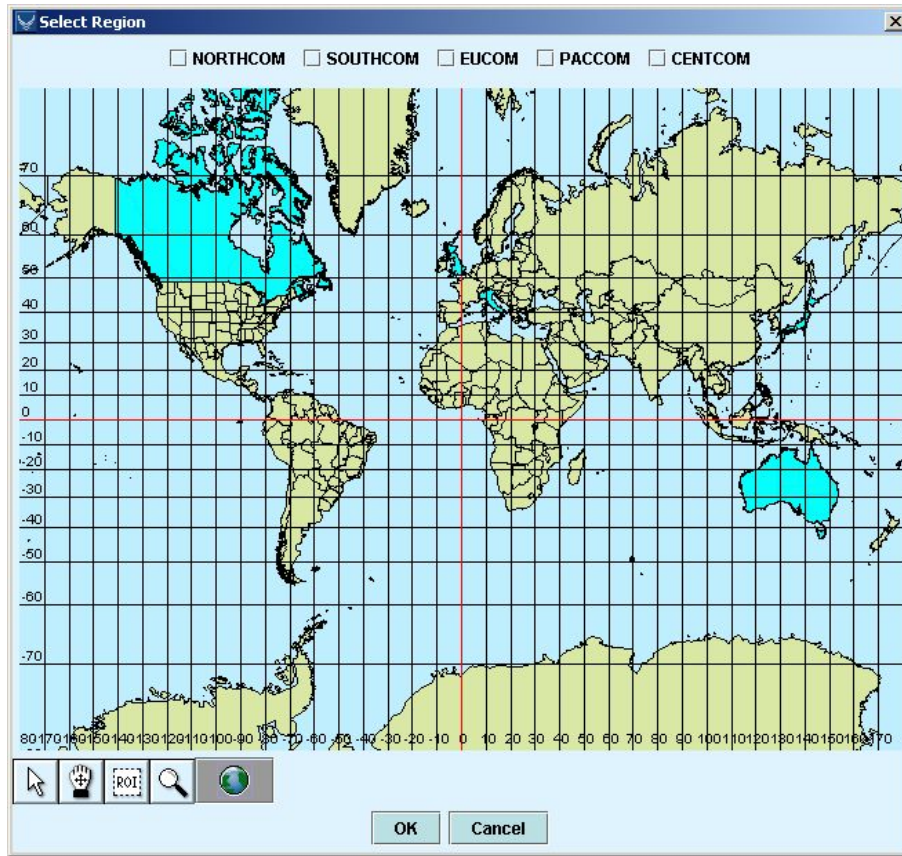


Figure 4.1 AGS Map Interface developed with GeoMap

4.3 Geo*View Integration

The plug-in implementation of the GeoMap toolkit allows users to perform visualization of geospatial imagery, visualize geospatial shape files, create annotations, add military symbols, and generate a resulting product in the NITF format. Each feature uses mouse driven tools that model themselves after those found in Geo*View. Only the symbology tool is new to Geo*View users. The symbol tool functions by first choosing a symbol using the accompanying icon browser pane and then clicking on the map where the center of the symbol should be placed. Once the symbology has been added to the map,

To implement the symbology visualization, the Batik Project from the Apache Foundation was utilized. This package is a Java based Scalable Vector Graphics (SVG) library that was designed to address the growing popularity of Extensible Markup Language (XML) graphics. The Batik Library was implemented without having to modify any of its native code. More information can be found on the Apache web page (<http://xml.apache.org/batik>).

5 Conclusions

The Map-GPL project has not only been successful in providing a map based interface for the Geospatial Product Library through the AGS program but has also produced a reusable toolkit for creation of applications requiring simple mapping capabilities. The GeoMap toolkit can be reused in any Java based application requiring basic visual geospatial registration capabilities.

It should be noted that the GeoMap toolkit provides accurate geospatial registration with several limitations. There are no restrictions on the type of imagery that may be displayed if geospatial corner coordinates are provided in the metadata. The imagery itself may not have been georectified in a manner consistent with accurate display of individual pixels. If using the GeoMap tool to overlay two images from different data sets, any inferences derived from coincidental pixels should be verified through an understanding of the specific georectification of each image product. This verification may include the use of some other toolset designed for high definition geospatial comparisons.

5.1 Future Work

If more time could be invested into GeoMap development, the GeoMap/Geo*View integration component should be enhanced for a more usable interface. Although the integration component demonstrates all functional requirements set forth by the Map-GPL program, several of the demonstrative capabilities could be made more robust. A usability study of the GeoMap/Geo*View integration component would assist developers in minimizing user error introduced while using the plug-in.

Support for re-projecting shape files from one given projection to another projection could be added. Currently shape files that specify their vertices geodetically can be interpreted correctly. To overcome the aforementioned limitation, the GeoMap toolkit would have to support reading of projection files.

The current saving capability does not factor in preserving the original metadata of the imagery it is producing. Implementation of ICHIP as defined in the National Imagery Transmission Format (NITF) Compendium of Controlled Extensions, Volume 2 as a Support Data Extension (SDE) would overcome this limitation. Having this SDE implemented would make the GeoMap/Geo*View integration component very useful for product generation.

A transition from relying on OpenMap's mapping framework to one developed specifically for the needs of PAR's Geo*Ware could be made. Although OpenMap is a robust, open-source package with acceptable e-mail support there are some aspects of the framework that are more robust for the needs of the Map-GPL toolkit. These robust features come with a price of a larger memory footprint as well as time spent when trying to trace a problem through native OpenMap code. It should be noted that these

limitations were identified as a risk early on in the contract. The trade off for this risk was the amount of unknowns present when trying to develop a mapping framework independent of any related third party libraries or tools.

6 Recommendations

GeoMap should be considered a viable option in producing applications that require basic map visualization capabilities. GeoMap has the ability to open shape files and many image file formats. This makes GeoMap an ideal tool for creating simple interfaces requiring display of geospatially aware data.

AGS could benefit from the use of GeoMap as a visualization tool. Currently AGS users are required to launch an external viewer if they desire to view the image data from a queried product. GeoMap's visualization capabilities could be leveraged so users would not have to leave the AGS environment to visualize an image.

7 References

OpenMap API: <http://openmap.bbn.com/doc/api/>

Batik Package: <http://xml.apache.org/batik/>

Map-GPL Requirements Analysis Report (CDRL A003)

Map-GPL Software Development Plan (CDRL A004)

Map-GPL Software Design Description (CDRL A005)

Map-GPL Interface Control Document (CDRL A006)

NITF Compendium of Controlled Extensions Vol. 2